REMARKS

Reconsideration and allowance of the above-referenced application are respectfully requested. Claims 1 and 5 amended. Claims 1-7 are pending in the application.

The specification has been amended to correct minor informalities. In addition, claim 1 has been amended to correct an informality and to ensure proper antecedent basis.

The telephonic interview between Examiner Divecha and the undersigned on September 18, 2006 is acknowledged with appreciation. During the interview, the Examiner indicated that amending the claims to recite "or" as a recital of alternative (e.g., one of A or B) would not render the claims indefinite under 35 USC §112, second paragraph. Hence, claims 1 and 5 as amended recite that the first entry identifier specifies ***one of*** a transmitted work queue entry ***or*** a subsequently transmitted work queue entry, and that the second entry identifier specifies ***one of*** the first transmitted work queue entry ***or*** an entry having received a subsequent acknowledgement.

Hence, it is believed the foregoing amendments overcome the objection to the specification, and the rejection of claims 1-7, under 35 USC §112, first paragraph.

The §101 Rejection is Traversed

The rejection of claims under 35 USC 101 is respectfully traversed, as the Examiner has failed to establish a *prima facie* case that "the claimed invention as a whole is directed to solely an abstract idea or to manipulation of abstract ideas or does not produce a useful result," as required under the USPTO Examination Guidelines (see MPEP 2106.II.A. at page 2100-7, Rev. 3, Aug. 2005). Further, the Examiner has failed to "expressly state how the language of the claims has been interpreted to support the rejection." *Id.*

In addition, the Examiner has failed to "identify the features of the invention that would render the claimed subject matter statutory if recited in the claim", as recommended under MPEP 2106.IV.B. (page 2100-11, Rev. 3, Aug. 2005).

Moreover, the assertion that the claims "can be simply be implemented on a piece of paper" is without foundation, lacks any rational basis, and demonstrates a complete disregard for

the explicit claim language as interpreted in view of the specification, as required under the USPTO Examination Guidelines. Moreover, the Examiner's assertion demonstrates a complete disregard for the law as interpreted by the USPTO according to the USPTO Examination Guidelines, and violates the legal requirements of *AT&T Corp. v. Excel Communications*, 50 USPQ2d 1447 (Fed. Cir. 1999).

In fact, the independent claims 1 and 5 specify storing, by a ***channel adapter***, entries having first link fields for formation of *first linked list* specifying a transmit sequence of transmitted work queue entries, and second link fields for generating *a second linked list* specifying an acknowledgement sequence of the transmitted work queue entries, ***based on the detected acknowledgement thereof***. The specification describes that the channel adapter (e.g., HCA 12 of Fig. 1), is compliant with the InfiniBand™ Architecture Specification and "implemented in a manner that ensures that ***hardware resources are efficiently utilized***" (page 4, lines 15-16). Further, the specification describes at page 1, lines 22-24 that the InfiniBand™ Architecture Specification provides an implementation in ***hardware*** of the transport layer services present in existing networking protocols. As such, the claimed storing ***in a table***, ***detecting an acknowledgement***, and ***generating in the table*** require <u>specific operations to be performed in hardware by the channel adapter implemented in ***hardware***</u>.

Hence, the assertion that the claims "can be simply be implemented on a piece of paper" demonstrates a disregard for explicit claim limitations, and an unreasonable interpretation of the claims in view of the specification and the interpretation that those skilled in the art would reach with respect to implementing the ***channel adapter*** according to the InfiniBand™ Architecture Specification, as described by the specification.

Further, the specification describes that the first linked list and the second linked list generated by the claimed channel adapter enables the useful, concrete and tangible result of tracking whether transmitted data packets need to be resent for work queue entries awaiting acknowledgement (e.g., according to a protocol requiring acknowledgement), in an economical and efficient manner (see, e.g., page 2, lines 3-9 and 17-29; page 3, lines 13-14, page 9, lines 31-

34, page 11, lines 5-8)[1]. As noted by the PTO Examination Guidelines, "[a]n applicant may assert more than one practical application, but ***only one is necessary to satisfy the utility requirement.***" (MPEP 2106.II.A at 2100-7).

Moreover, the subject claims are statutory under *AT&T Corp. v. Excel Communications*, 50 USPQ2d 1447 (Fed. Cir. 1999), where the Federal Circuit held statutory claims directed to a method comprising generating a message record for an interexchange call, and "including, in said message record, a primary interexchange carrier (PIC) indicator having a value which is a function of whether or not the interexchange carrier ... is a predetermined one of said interexchange carriers" because "[t]he PIC indicator represents information about the call recipient's PC, a useful, non-abstract result that facilitates differential billing of long-distance calls made by an [interexchange carrier's] subscriber. *Id* at 1452.

For these and other reasons, the §101 rejection of claims 1-7 must be withdrawn.

The §103 Rejection

Claims 1-7 stand rejected under 35 USC 103 in view of U.S. Patent No. 6,735,642 to Kagan et al. in view of U.S. Patent No. 6,611,883 to Avery. This rejection is respectfully traversed.

Each of the independent claims 1 and 5 specify storing, in a table, entries identifying ***transmitted*** work queue entries: each work queue entry also includes a first link field storing first entry identifiers, wherein the first entry identifiers in the respective first link fields of the multiple entries form a first linked list specifying a ***transmit sequence*** of the ***transmitted work queue entries***. Hence, each of the independent claims 1 and 5 specify that the work queue entry is added to the table based on ***transmission of the corresponding work queue entry***.

Each work queue entry also includes a ***second*** link field for storing a second entry identifier, the second link fields forming a second linked list specifying an acknowledgement

---

[1]In fact, this feature is explicitly recited in claims 3 and 6.

sequence of the transmitted work queue entries, based on an acknowledgement of at least a first of the transmitted work queue entries stored in the table.

Hence, the claimed channel adapter generates in the table ***two linked lists***, namely the first linked list specifying a *transmit sequence* of the transmitted work queue entries, and the second linked list specifying the acknowledgement sequence of the transmitted work queue entries. These and other features are neither disclosed nor suggested in the applied prior art.

No Disclosure or Suggestion in Kagan et al. of Entries Identifying *Transmitted* Work Queue Entries

Kagan et al provides no disclosure or suggestion of storing an entry having a work queue entry field that specifies a ***transmitted work queue entry***, where the entry is added based on ***transmission*** of the work queue entry, as claimed. Rather, Kagan describes the storage in system memory (26 of Fig. 1) of a chain 40 of DMA descriptors 42 (Fig. 2) that are "prepared in memory 26 by a **program running on CPU 28**"(col. 7, lines 29-31). In other words, the disclosed chain 40 does not teach a ***transmitted*** work queue entry, rather the disclosed chain 40 represents a chain of descriptors generated by a program running on CPU 28 and that are **to be transmitted** by the DMA engine 22:

> ***A program running on CPU 28, typically an application software program, generates a chain of descriptors for execution by DMA engine 22***. The CPU stores the descriptor chain in memory 26 and then notifies the DMA engine that the chain is ***ready for execution*** by ringing a doorbell of the [DMA] engine.
>
> (Col. 6, lines 54-59; see also col. 7, lines 29-31 and col. 8, lines 1-12).

The program running on the CPU 28 in Kagan et al. notifies the DMA engine 22 of the updated descriptors in memory 26 by "ringing a doorbell", namely by setting a "Last Executed" field in the control register 30 identifying the location of the next descriptor in system memory 26 to be executed (col. 6, line 61 to col. 7, line 7; col. 8, lines 13-16). The DMA engine 22

responds to the doorbell by ***reading the descriptor in system memory 26*** at the location specified in the "Last Executed" field in the control register, processing the descriptor, updating the corresponding status field 54 after execution (from "0" to "1"), and fetching the next descriptor specified in the corresponding field 50 if the next address is not a null address (col. 6, line 61 to col. 7, line 4; col. 7, lines 14-22 and 51-67; col. 8, lines 16-19 and line 31 to col. 9, line 1).

Hence, the DMA engine 22 does not transmit the data associated with the descriptor until after ***reading*** the descriptor ("the [DMA] engine transfers data ***from memory 26 to a target device 36***", col. 7, lines 17-18).

Moreover, the DMA engine 22 performs no storage or writing in the memory 26 that stores the chain 40 of descriptors 42, *except* to update the status field 54 after execution of the corresponding descriptor (col. 7, lines 61-65; col. 8, lines 64-67). The status field 54 is used by the CPU 26 to append a new DMA descriptor chain onto the end of the preceding chain if the status field indicates a completed final descriptor (e.g. "2") (col. 7, lines 62-67; col. 9, lines 7-25), or by system memory management that ***reclaims the completed nonfinal descriptor*** if the status field indicates a completed non-final descriptor (col. 9, lines 1-5).

Note, however that the updating of the status field 54 by the DMA engine 22 is not a teaching of the claimed storing (adding) of a table entry identifying a ***transmitted work queue entry***, especially since the status field 50 is not a link field configured for referencing another entry in the table, as claimed.

Hence, Kagan et al. provides no disclosure or suggestion of storing (adding) an entry having a work queue entry field that specifies a ***transmitted work queue entry***, as claimed.

No Disclosure or Suggestion in Kagan et al. of the Claimed Second Link Field

Kagan et al. also provides no disclosure or suggestion of the claimed feature that each entry in the table *also* includes a second link field configured for referencing another entry in the table, as asserted by the Examiner. In fact, Fig. 2 of Kagan et al. consists of only a single link field, namely the "Next" field 50; the remaining fields 44, 46, 48, and 54 are not teachings of the claimed second link fields because they specify values (e.g., Op Code, Source Address,

Destination Address, Status), and not another entry in the table, as required in the claims (see col. 7, lines 29-67).

Hence, Kagan et al. does not teach or suggest that each claimed entry includes a ***second link field***, as asserted by the Examiner.

Avery et al.

Applicant traverses the June 19, 2006 Office Action as incomplete because it fails to address the arguments regarding Avery et al. as presented in the Appeal Brief filed March 27, 2006. Specifically, the Office Action specifies that prosecution is reopened in view of the Appeal Brief, but fails to address Applicant's arguments as specified in the Appeal Brief as applied to the current rejection under §103. Rather, the rejection asserts that Avery teaches the claimed second linked list specifying an acknowledgement sequence of the transmitted work queue entries, with no comments on the Arguments on pages 10-12 of the Appeal Brief demonstrating that Avery does *not* disclose or suggest the claimed second linked list.

Hence, Applicant traverses the Official Action as incomplete because it fails to answer the material traversed. See MPEP §707.07(f).

Avery does not disclose or suggest a second linked list ***specifying the acknowledgement sequence***, as claimed. In fact, this rejection under §103 fails to identify *which* portion of Avery should be considered a teaching of the claimed second linked list.

The Final rejection in the October 26, 2005 Office Action had erroneously assumed that the pointer 715 to the DMA context scoreboard constitutes a "second linked list specifying an acknowledgement sequence". However, Avery explicitly specifies that the DMA scoreboard 770 keeps tracks of outstanding DMA requests based on the size of the data transfer or the number of prefetches.

Specifically, Avery specifies that the DMA scoreboard 770 "is a data structure that holds the DMA context and tracks outstanding DMA requests to insure that all outstanding requests are completed. The DMA scoreboard 770 contains data tags 774 to track data returning from

outstanding DMA read requests and ACK flags which track whether DMA writes have been completed." (Col. 9, lines 59-65).

Further, the DMA scoreboard 770 includes a "prefetch section that includes **the number of prefetches** 776 and a **size section** 775 (col. 9, line 67 to col. 10, line 1). Avery describes prefetching with reference to Figs. 7 and 9B by increasing a prefetching address by the prefetching size 774 until the DMA transfer size 775 is equal to the work queue entry size request 710:

> Since these [DMA read] requests are acted on asynchronously by the InfiniBand℠ bus system, they must be coordinated by the DMA context scoreboard 770 ***which stores the number and size of additional read requests that must be made in fields 774 and 775, respectively***.
>
> (Col. 10, line 65 to col. 11, line 3)
>
> A tag that uniquely identifies the request is stored in field 774. ... The data transfer software then monitors, in step 922, the returning data packets until a response packet corresponding to the outstanding request is received. The response packet can be examined to determine whether it corresponds to the initial read request.
>
> A prefetch is performed using the same work queue pair that was used to perform the initial DMA read. When a response packet is received from the initial DMA read request, the process then proceeds to step 924 where the DMA scoreboard is updated by adding the prefetch size stored in field 774 to the current address and storing the result in next address field 772. ***A check is then made in step 926 to determine whether the DMA transfer size is equal to the work queue entry size request 710. If the sizes are equal, the process finishes in step 928.***
>
> If prefetch data remains to be retrieved, the next prefetch request is initiated in step 930 by combining the next address in DMA scoreboard field 772 with the R-Key 764 in the address map 754 as schematically indicated by block 760 to produce a new virtual address which is pushed into work queue entry 703. The process then returns to step 920 in which the new prefetch read request is generated. ***Operation continues in this manner until the size 710 in the work queue entry 703 is exhausted as determined in step 926*** or, alternatively, until internal buffer resources are consumed.
>
> (Col. 11, lines 7-36).

Hence, Avery specifies that, in response to an acknowledgement that matches the acknowledgement tag 776, the next address 770 is incremented by the prefetch size 774, and a new fetch address is created at block 760 based on combining the next address 770 with the R-Key 708, until the DMA transfer size 775 is equal to the work queue entry size request 710.

Hence, Avery provides no disclosure or suggestion of storing an ***acknowledgement sequence*** of transmitted work queue entries, as claimed; rather, Avery simply determines whether an acknowledgement for a specific data packet has been received, and continues to output additional packets until the DMA transfer size 775 equals the work queue entry size request 710.

Hence, there is no disclosure or suggestion of the claimed first linked list ***specifying a transmit sequence*** of the transmitted work queue entries generated from first link fields, in combination with a second linked list ***specifying an acknowledgement sequence of the transmitted work queue entries*** generated from second link fields, as claimed.

Hypothetical Combination of Kagan et al. and Avery

The hypothetical combination of Kagan et al. and Avery would neither disclose nor suggest that ***a single table*** would have entries, where ***each entry includes a first link field*** (for a first linked list specifying a transmit sequence of transmitted work queue entries) ***and a second link field*** (for a second linked list specifying an acknowledgement sequence of the transmitted work queue entries), especially because Avery teaches away from the claimed combination of first and second link fields for creation of respective first and second linked lists ***in a single table***.

Rather, the Examiner's reliance on col. 3, lines 18-25 and col. 9, lines 58-67 (describing the DMA scoreboard) as a motivation to modify Kagan et al. and Avery belies the fact that Avery stresses *repeatedly* that the DMA scoreboard is a special data structure that is ***distinct*** from the InfiniBand Address map 754. For example, the Abstract at lines 1-4 specifies that:

> Speculative prefetching during DMA reads in a message-passing, queue-oriented bus system is controlled by creating a special data structure, called a "DMA scoreboard", for each work queue entry associated with a DMA read.

(See also column 3, lines 18-20: "speculative prefetching is controlled by **creating a special data structure**, called a "DMA scoreboard", for each work queue entry associated with a DMA read with prefetching enabled.").

Further, Avery identifies the InfiniBand Address map 754 as a ***distinct data structure***:

> The InfiniBand address map 754 is a data structure that is stored locally in the InfiniBand-PCI bridge 324 and ***has a plurality of entries of which entries 756 and 758 are shown***. Each entry is associated with a region in the PCI address space 720 and holds the initial segment address for each region that is mapped into the system virtual memory address space, through the host channel adapter TPT.
>
> (Col. 8, lines 57-60).

Avery also suggests that the work queue pairs should be distinct data structures:

> HCA 308 [of Fig. 3] has a work queue pair consisting of send queue 310 and receive queue 312. Similarly, TCA 324 has a work queue pair consisting of send queue 326 and receive queue 328. Although only two queue pairs are shown, typically each client would create many more work queue pairs in order to conduct its operation. In order to use the work queue pair, a client submits a work request to its respective channel adapter and the work request causes an instruction called a Work Queue Entry (WQE) to be placed on the appropriate send work queue.
>
> (Col. 6, lines 21-30).

Finally, Avery describes the DMA scoreboard 770 as a distinct data structure:

> The work queue entry 703 also contains a pointer 715 to a DMA context scoreboard 770. ***The DMA scoreboard 770 is a data structure*** that holds the DMA context and tracks outstanding DMA requests to insure that all outstanding requests are completed.
>
> (Col. 9, lines 58-62).

Based on the foregoing, there is no disclosure or suggestion whatsoever that the InfiniBand Address Map 754, the RDMA Work Queue Pair (consisting of work queue entries 703 and 705), and the DMA context scoreboard 770 would be implemented within the same data structure, especially since each of the Address Map 754, the work queue entries 703 and 705, and the DMA context scoreboard 770 have distinct data field types that have no similarly among each other.

Hence, the assertion that the address map 754, the work queue entries 703 and 705, and the DMA context scoreboard 770 should be in the same data structure has no rational basis, and is inconsistent with the explicit teachings of Avery.

Hence, there is no disclosure or suggestion that one skilled in the art would have been motivated to modify Kagan et al., as described in Avery, to result in a hypothetical combination where ***a single table included first and second link fields for first and second linked lists specifying a transmit sequence and an acknowledgement sequence, respectively***. Although the test for establishing an implicit motivation in the prior art is what a prior art statement would have suggested to those of ordinary skill, such a statement "must be considered ***in the context of the teaching of the entire reference***." *In re Kotzab*, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000).

For these and other reasons, the §103 rejection must be withdrawn.

In view of the above, it is believed this application is in condition for allowance, and such a Notice is respectfully solicited.

To the extent necessary, Applicant petitions for an extension of time under 37 C.F.R. 1.136. Please charge any shortage in fees due in connection with the filing of this paper, including any missing or insufficient fees under 37 C.F.R. 1.17(a) or 1.17(p), to Deposit Account No. 50-0687, under Order No. 95-507, and please credit any excess fees to such deposit account.

Respectfully submitted,
Manelli Denison & Selter PLLC

Leon R. Turkevich
Registration No. 34,035

Customer No. 20736
Date: September 19, 2006